The Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG) Shared Task 2021

Don Tuggener **Ahmad Aghaebrahimian**
Zurich University of Applied Sciences (ZHAW)
Winterthur, Switzerland
{tuge, agha}@zhaw.ch

Abstract

This paper describes the first Sentence End and Punctuation Prediction in Natural Language Generation (SEPP-NLG) shared task[1] held at the SwissText conference 2021. The goal of the shared task was to develop solutions for the identification of sentence boundaries and the insertion of punctuation marks into texts produced by NLG systems. The data and submissions[2], and the codebase[3] for the shared tasks are publicly available.

1 Introduction

Sentence End Detection, also known as Sentence boundary disambiguation (SBD) or boundary detection, is the Natural Language Processing (NLP) task of recognizing where a sentence begins and ends. A period is the most common end of sentence indicator in written English as well as many other Indo-European languages. However, a period may be used in a decimal point, an abbreviation, an email address, or other possible cases as well which makes sentence boundary detection a challenge. Other forms of punctuation such as question and exclamation marks, semicolons, comma, etc. add to this challenge. Although sentence boundary detection is considered an almost solved issue for formal written language (Walker et al., 2001), it poses a challenge in terms of meaning distortion and readability in synthetic or automatically translated or transcribed texts such as the output of Automatic Speech Recognition (ASR) or Machine Translation (MT) systems. The punctuation marks in such synthetic text may be displaced for several reasons. Detecting the end of a sentence and placing an appropriate punctuation mark improves the quality of such texts not only by preserving the original meaning but also by enhancing their readability.

The goal of the SEPP-NLG shared task is to build models for identifying the end of a sentence by detecting an appropriate position for putting an appropriate punctuation mark.

2 Related Work

Similar to the system proposed by Grefenstette and Tapanainen (1997), the earliest attempts for sentence boundary detection utilize a set of rules or regular expressions. In a different direction, Reynar and Ratnaparkhi (1997), and Kiss and Strunk (2006) proposed an information-centric approach based on the Maximum Entropy model, and an unsupervised method based on collocation statistics respectively. Decision tree classifier (Riley, 1989), Naïve Bayes (López and Pardo, 2015) and deep learning based (Kaur and Singh, 2019) models are the most recent advances based on machine learning that are proposed for predicting correct positions for the period in particular and other punctuation marks in general. Moving forward and combining the rule-based and machine learning-based systems, Deepamala and Ramakanth (2012) proposed a hybrid system with high performance.

Our task is closely related to Tilk and Alumäe (2016) and follow-up work that uses the Europarl and TED talk corpora for punctuation prediction. Similar to our goal, Żelasko et al. (2018); Donabauer et al. (2021) investigate sentence boundary detection in unpunctuated ASR outputs of spo-

[1] https://sites.google.com/view/sentence-segmentation/

[2] https://drive.switch.ch/index.php/s/g3fMhMZU2uo32mf

[3] https://github.com/dtuggener/SEPP-NLG-2021

ken dialogues based on textual features. Cho et al. (2017) propose a method to predict sentence boundaries and punctuation insertion in a real-time spoken language translation tool. In a similar setting, Klejch et al. (2017) include acoustic features to improve punctuation prediction in a speech translation system, and Yi and Tao (2019) combine lexical and speech features for punctuation prediction in a traditional ASR setting. Finally, Rehbein et al. (2020) investigate the annotation and detection of sentence like units in spoken language transcripts.

3 Task Overview

Ultimately, the goal of SEPP-NLG is to predict sentence ends and punctuation in NLG texts. However, there are no corpora that feature NLG texts and their manually transcribed and corrected versions. Therefore, we approximate the setting by using a) transcripts of spoken texts, and b) lower-casing the texts and removing all punctuation marks. While there are multiple corpora of transcribed spoken language, we choose the Europarl corpus[4] (Koehn, 2005) as the source for our data. The Europarl corpus consists of transcripts of the sessions of the European parliament and features transcripts in multiple languages.

We offer the following subtasks:

- **Subtask 1:** (fully unpunctuated sentences-full stop detection): Given the textual content of an utterance where all punctuation marks are removed, correctly detect the end of sentences by placing a full stop in appropriate positions.

- **Subtask 2:** (fully unpunctuated sentences-full punctuation marks): Given the textual content of an utterance where all punctuation marks are removed, correctly predict all punctuation marks.

Participants were free to choose for which languages and subtasks they contributed a submission, but were encouraged to participate in all languages.

3.1 Data

We leverage the open parallel corpus (OPUS) version of the Europarl corpus[5] (Tiedemann, 2012) for extracting the task data as it provides sentence boundaries and tokenization. Albeit the sentence boundaries in the corpus are automatically generated, they are quite reliable as the data and the models trained to detect the boundaries contain all the original punctuation symbols of the transcripts.

In the spirit of the "Swissness" of the SwissText conference where SEPP-NLG 2021 is co-located, we select 3 of the 4 official languages[6] of Switzerland, i.e. German, French, and Italian and complement the selection by incorporating English.[7]

The Europarl corpus contains multiple punctuation symbols. For subtask 2, we gauged which subset of them represents a realistic and feasible goal for their automatic prediction in a stream of unpunctuated, lower-cased tokens. Also, we considered which punctuation marks improve the readability of a text the most. Hence, we consolidated the selection of punctuation symbols for subtask 2 to $: -, ?.0$ (0 indicating no punctuation) and mapped the symbols !; to ., the period. We removed all sentences from the data that contain other punctuation symbols such as parentheses, as there is no straightforward way to remove punctuation without interfering with the naturalness of a sentence. This removal affected the data for both subtasks and resulted in removing less than 10% of the data per language. We also removed HTML artifacts, and special (non-visible) characters (zero width space, soft hyphen) from the data. Finally, we omitted sentences with fewer than 3 tokens and documents with fewer than 2 sentences.

The data format is as follows: Lower-cased tokens per file are listed vertically, and the labels for subtask 1 (binary classification) and 2 (multiclass classification) are appended horizontally, separated by tab. The labels encode whether a token emits a sentence end (subtask 1) and a punctuation symbol (subtask 2). Table 1 shows an example.

Per language, we randomly selected 80% of the documents for the training and 20% for the test set. From the the training set, we then randomly sampled 20% of the documents as the development set.

Table 2 shows several statistics of our data. We see similar properties for all languages: Most sentences are unique, and there are few sentences that occur both in the train and test sets.[8] German fea-

[4] http://www.statmt.org/europarl/

[5] https://opus.nlpl.eu/Europarl.php

[6] The forth, Romansh, is not represented in Europarl.

[7] Incorporating further languages from the OPUS corpus using our scripts is seamless as the data format is consistent across languages.

[8] Duplicate sentences are often formulaic, administrative ones, like "The session is adjourned." etc.

Token	Label 1	Label 2
the	0	0
next	0	0
item	0	0
is	0	0
the	0	0
commission	0	0
statement	0	0
on	0	0
the	0	0
referendum	0	0
in	0	0
venezuela	1	.
member	0	0
of	0	0
the	0	0
commission	0	.
madam	0	0
president	0	,
the	0	0

Table 1: Example of the data format.

tures the largest vocabulary, as is expected due to its morphological richness, and the vocabulary overlap between train and test sets is roughly 50% for all languages.

Concerning the labels, the data is highly skewed towards the 0 label for both tasks, as most tokens do not emit a sentence end or punctuation symbol after them. For example, there are 9'618'776 tokens with the label 0 and 420'446 with label 1 subtask one in the English test set, which yields an average sentence length of almost 24 tokens. Table 3 shows a breakdown of the label counts in the English test set for subtask 2. It shows that the period and comma symbols have similar counts and are the most frequent labels among the non-0 labels. The remaining labels occur less than an order of magnitude less frequently. These label distribution properties are similar across all languages.

3.2 Surprise Test Data

The Europarl corpus covers domain-specific language, i.e. political statements in the European parliament. To measure how well the participating systems trained on our data generalize to out-of-domain data, we incorporated a surprise test set comprised of TED talk transcripts[9] (Reimers and Gurevych, 2020).

For each language, we sampled 500 TED talks, favoring those that have the lowest vocabulary overlap with our Europarl test sets to maximize the vocabulary shift. The document-based average percentage of the vocabulary overlap ranges from 85 to 90, meaning there are on average 10-15% of tokens per document in the surprise test set that are not in the Europarl test set.

While being one order of magnitude smaller than the Europarl test set, the surprise test set is also highly and similarly imbalanced regarding the label distribution. In the English surprise test set, there are 67'446 tokens with label 1 and 1'014'464 tokens with label 0. This yields an average sentence length of 16 tokens, which is significantly lower than the 24 tokens in the English Europarl test set. The label counts for subtask 2 follow an almost identical distribution in both test sets.

4 Submissions

ZHAW-mbert: We provided a baseline based on the multilingual BERT model (Devlin et al., 2019), mBERT, implemented in the simpletransformers library[10]. We treat the task as a token classification problem and segment the documents into subsequent, non-overlapping chunks of length 512 to adhere to the sequence length restrictions of BERT. We fine-tuned the model on the training data of all languages with a randomly shuffled file order across all languages and vanilla settings for about one week on a single GPU.

ZHAW-adapter-mbert: To contrast the resource-intensive fine-tuning of mBERT with a computationally cheaper approach of task adaption, we apply the adapter-transformers library[11] (Pfeiffer et al., 2020). Instead of updating all the weights of the base models (mBERT in our case), the adapters approach inserts a few feed-forward layer in between the transformer blocks and only trains those for adapting a base model to a new task. We again use the vanilla settings and train the model for one day.

OnPoint: In their study of sentence segmentation, Michail et al. (2021) proposed a majority-voting ensemble model consisting of several Transformer models trained in different ways. The models' predictions are leveraged at test time using a sliding window to obtain the final predictions. They offered their system as language-dependent models for all four languages of the shared task and both sub-tasks.

[9] https://opus.nlpl.eu/TED2020.php

[10] https://github.com/ThilinaRajapakse/simpletransformers

[11] https://github.com/Adapter-Hub/adapter-transformers/

Lang	#sentences	unique	train∩test	#tokens	unique	train∩test
EN	1'406'577	1'382'738	2'660	33'779'095	88'370	43'744
DE	1'308'508	1'276'691	2'806	28'645'112	294'035	112'000
FR	1'236'504	1'215'981	2'081	32'690'367	103'774	57'112
IT	1'132'554	1'112'742	1'746	28'167'993	131'024	67'626

Table 2: Training data statistics, showing number of (unique) sentences and tokens and the number of sentences and tokens in both training and test set (train∩test) per language.

Label	Count
0	9'050'256
,	521'594
.	417'560
-	23'600
:	13'146
?	13'066

Table 3: Label distribution for subtask 2 in the English test set.

Unbabel-INESC-ID: Rei et al. (2021) extend the architecture proposed by Rei et al. (2020) to develop a multilingual model for sentence end and punctuation prediction. Their system is designed based on pre-trained contextual embeddings and built on top of a pre-trained Transformer-based encoder model. They propose their method as a single multilingual model for all languages and subtasks of the shared task.

UR-mSBD: Donabauer and Kruschwitz (2021) propose a system based on a pre-trained BERT model and fine-tuned for the first sub-task. They use language-specific models for each of the four languages of the shard task. They consider sub-task 1 as a binary classification problem by identifying tokens that indicate the position of a full stop.

oneNLP: Applying multi-task Albert for English and multi-lingual Bert for other languages Mujadia et al. (2021) explored the impact of using contextual language models for sentence end and punctuation prediction. They modeled the problem in both subtasks as a sequence labeling task. They presented the results of employing a baseline CRF, as well as the results of applying a fine-tuning method over contextual embedding.

HULAT_UC3M: Based on the Punctuator framework (Tilk and Alumäe, 2016) which is a bidirectional recurrent neural network model equipped with an attention mechanism, Masiello-Ruiz et al. (2021) developed an automatic punctuation system named HULAT-UC3M. They trained HULAT-UC3M for all languages as well as both sub-tasks in the shared task individually.

HTW: Guhr et al. (2021) modeled the task as a token-wise prediction and examined several language models based on the transformer architecture. They trained two separate models for the two tasks and submitted their results for all four languages of the shared task. They advocated transfer learning for solving the task and showed that the multilingual transformer models yielded better results than monolingual models. By pruning the Bert layers, they also showed that their model retains 99% of its performance without 1/4 of the last layers.

5 Results

In section 3.1 we showed that our data is highly imbalanced regarding the label distribution. Accuracy or Macro F1 scores are not suitable metrics in this setting, as majority class prediction would yield an accuracy of 96% for subtask 1 on the English test set, e.g. Therefore, we applied the following metrics to evaluate the participants' submissions:

- **Subtask 1:** F1 score of the label 1 (the positive class, i.e. sentence end)
- **Subtask 2:** Macro F1 of the selected punctuation symbols

We observe that a) most systems achieve a very high score for subtask 1 for all languages on the Europarl data, and b) the F1 scores are almost identical (with seemingly minor differences in precision and recall) for the top-ranking systems for both tasks. Further, the top-ranking systems are the same ones for both tasks. This is to be expected to some degree, as it can be argued that subtask 2 subsumes subtask 1.

While the F1 scores for subtask 2 seem low compared to subtask 1, a more detailed results analysis reveals that the lower (Macro) F1 scores mainly stem from the labels with the lowest counts in the data. Table 6 gives the detailed classification report

	EN			DE			FR			IT			AVG		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
TEST SET															
htw+t2k_fullstop_multilang	0.94	0.95	0.94	0.95	0.96	0.96	0.94	0.94	0.94	0.92	0.94	0.93	0.94	0.95	0.94
OnPoint	0.93	0.95	0.94	0.95	0.96	0.96	0.92	0.94	0.93	0.90	0.95	0.92	0.93	0.95	0.94
Unbabel-INESC-ID	0.94	0.94	0.94	0.95	0.96	0.96	0.94	0.94	0.94	0.92	0.94	0.93	0.94	0.95	0.94
UR-mSBD	0.91	0.92	0.92	0.94	0.96	0.95	0.93	0.94	0.93	0.91	0.93	0.92	0.92	0.94	0.93
ZHAW-mbert	0.91	0.93	0.92	0.93	0.96	0.95	0.90	0.93	0.91	0.88	0.93	0.90	0.91	0.94	0.92
oneNLP	0.92	0.92	0.92	0.93	0.95	0.94	0.90	0.89	0.89	0.88	0.89	0.89	0.91	0.91	0.91
ZHAW-adapter-mbert	0.88	0.90	0.89	0.79	0.85	0.82	0.81	0.84	0.83	0.77	0.78	0.77	0.81	0.84	0.83
HULAT_UC3M	0.86	0.80	0.83	0.23	0.90	0.36	0.86	0.79	0.83	0.84	0.78	0.81	0.70	0.82	0.71
htw+t2k_fullstop_german				0.95	0.96	0.95									
SURPRISE TEST SET															
htw+t2k_fullstop_multilang	0.85	0.70	0.77	0.90	0.74	0.82	0.84	0.70	0.76	0.85	0.67	0.75	0.86	0.70	0.78
OnPoint	0.84	0.75	0.80	0.89	0.77	0.82	0.82	0.72	0.77	0.83	0.71	0.77	0.85	0.74	0.79
Unbabel-INESC-ID	0.92	0.75	0.83	0.88	0.71	0.78	0.85	0.72	0.78	0.86	0.68	0.76	0.88	0.72	0.79
UR-mSBD	0.82	0.68	0.74	0.89	0.73	0.80	0.83	0.70	0.76	0.84	0.67	0.74	0.85	0.70	0.76
ZHAW-mbert	0.78	0.70	0.74	0.86	0.74	0.80	0.78	0.69	0.73	0.77	0.65	0.70	0.80	0.70	0.74
oneNLP	0.81	0.67	0.73	0.85	0.72	0.78	0.77	0.62	0.69	0.78	0.58	0.67	0.80	0.65	0.72
ZHAW-adapter-mbert	0.75	0.69	0.71	0.75	0.69	0.72	0.72	0.67	0.69	0.71	0.55	0.62	0.73	0.65	0.69
HULAT_UC3M	0.68	0.41	0.51	0.41	0.61	0.49	0.74	0.41	0.53	0.73	0.30	0.43	0.64	0.43	0.49
htw+t2k_fullstop_german				0.90	0.75	0.80									

Table 4: Results for subtask 1

	EN			DE			FR			IT			AVG		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
TEST SET															
htw+t2k_fullstop_multilang	0.82	0.74	0.77	0.84	0.79	0.81	0.83	0.75	0.78	0.82	0.72	0.76	0.83	0.75	0.78
OnPoint	0.81	0.75	0.77	0.82	0.80	0.81	0.78	0.77	0.77	0.77	0.74	0.75	0.80	0.77	0.78
Unbabel-INESC-ID	0.83	0.72	0.76	0.84	0.77	0.80	0.83	0.74	0.77	0.82	0.70	0.74	0.83	0.73	0.77
ZHAW-mbert	0.80	0.71	0.74	0.82	0.75	0.78	0.81	0.71	0.75	0.79	0.66	0.71	0.81	0.71	0.75
oneNLP	0.79	0.69	0.72	0.80	0.74	0.77	0.79	0.65	0.68	0.78	0.62	0.66	0.79	0.68	0.71
HULAT_UC3M	0.76	0.60	0.63	0.79	0.65	0.69	0.75	0.59	0.64	0.71	0.52	0.57	0.75	0.59	0.63
ZHAW-adapter-mbert	0.78	0.64	0.68	0.59	0.48	0.49	0.70	0.55	0.59	0.64	0.46	0.49	0.68	0.53	0.56
SURPRISE TEST SET															
htw+t2k_fullstop_multilang	0.65	0.57	0.60	0.68	0.64	0.66	0.66	0.60	0.62	0.61	0.53	0.56	0.65	0.59	0.61
OnPoint	0.65	0.59	0.62	0.66	0.65	0.65	0.63	0.60	0.61	0.57	0.55	0.56	0.63	0.60	0.61
Unbabel-INESC-ID	0.68	0.57	0.61	0.71	0.63	0.65	0.69	0.59	0.63	0.63	0.53	0.56	0.68	0.58	0.61
ZHAW-mbert	0.62	0.51	0.55	0.66	0.58	0.60	0.64	0.54	0.57	0.51	0.45	0.47	0.61	0.52	0.55
oneNLP	0.62	0.52	0.56	0.61	0.57	0.58	0.61	0.48	0.51	0.54	0.43	0.46	0.60	0.50	0.53
HULAT_UC3M	0.50	0.40	0.43	0.59	0.47	0.51	0.56	0.38	0.41	0.45	0.33	0.36	0.53	0.40	0.43
ZHAW-adapter-mbert	0.60	0.48	0.51	0.54	0.41	0.44	0.60	0.44	0.48	0.51	0.35	0.38	0.56	0.42	0.45

Table 5: Results for subtask 2

for the top three ranking system for the English test set. It shows that the systems are able to predict periods, commas, and question marks reliably, but that they struggle with hyphens and colons, which lowers the Macro F1 scores.

Label	htw+t2k	OnPoint	Unbabel
0	0.99	0.99	0.99
,	0.82	0.82	0.80
.	0.95	0.95	0.94
-	0.42	0.41	0.37
:	0.57	0.57	0.56
?	0.88	0.91	0.89

Table 6: F1 scores per label for the top-performing systems on the English test set for subtask 2.

All systems perform significantly worse on the surprise test sets for both tasks. To gauge the difficulty of the task on the TED dataset compared to the Europarl dataset, we train the ZHAW-mbert approach on the remaining TED talks that were not selected for the surprise test set and then test the system on the surprise test set. Table 7 shows that the average F1 score does improve by 11 percentage points when training the ZHAW-mbert system on domain data. Still, the 0.66 F1 score is 9 percentage points behind the average F1 score on the Europarl data. Hence, the drop in performance of Europarl-trained ZHAW-mbert on the surprise test set can both be accounted for by the domain shift and by the increased difficulty of the target domain (TED talks). We expect that this applies for the performance drop of all systems.

	Prec.	Rec.	F1
ZHAW-mbert	0.76	0.63	0.66

Table 7: Results of training ZHAW-mbert on TED talks for subtask 2 (averaged over all languages).

We expected some submissions to use linguistic features such as part-of-speech tags or partial syntax parse trees and hypothesized that such systems would fare better on out-of-domain data. However, all participating systems applied neural encodings of the surface tokens and did not encode linguistic features explicitly. Still, the ranking of the systems remains intact on the surprise test sets.

The top three systems in both tasks all use transformers-based approaches and tackle the tasks in a similar manner. We hypothesize that this is the main reason for near identical performance of the systems in terms of F1 scores. Based on the task results, these three systems seem to produce near-identical output. To better gauge their similarities and differences, we evaluate their outputs for subtask 2 in a pair-wise manner on the English test set. We apply the evaluation metric such that one system output takes the role of the ground truth and the other the one of the system prediction, which yields the F1 scores per class that we leverage as an indicator of the similarity or agreement of the per-token predictions. Table 8 shows the results. While the macro F1 scores and even the per-class F1 scores in Table 6 are highly similar, there are significant differences in this analysis. For example, for the hyphen class, the systems have different predictions in over 30% of the cases, and for colon in roughly 20%. For the majority classes of the non-0 classes, the systems disagree in about 10% of the cases for comma, but their predictions are highly similar for period (96% agreement).

Label	htw+t2k vs Unbabel	OnPoint vs Unbabel	OnPoint vs htw+t2k
0	0.99	0.99	1.00
,	0.90	0.90	0.92
.	0.96	0.96	0.96
-	0.67	0.66	0.68
:	0.79	0.81	0.81
?	0.89	0.92	0.91

Table 8: System prediction similarity between the three top-performing systems on the English test set for subtask 2.

Following Tuggener (2017), we can take the comparison a step further and analyse the type of differences per label. For example, the OnPoint submission's F1 score for hyphen is 4 percentage points higher than the one of Unbabel, and their prediction agreement for hyphen is 68%. This does not indicate, however, whether OnPoint's predictions are always better. The aforementioned comparison takes a ground truth label G, the predicted label A of one system, and the predicted label B of another system and defines three types of differences for the cases where $A \neq B$:

- correction: $G = B$
- new error: $G = A$
- changed error: $G \neq A \neq B$

Table 9 shows the results. We see that the predictions of commas makes up a large portion of the differences. When OnPoint's prediction differs from Unbabel's for comma, OnPoint is correct and Unbabel incorrect in nearly 70% of the cases, which explains the 2 percentage point higher performance of OnPoint in Table 6. Still, Unbabel is correct in almost 30% of the cases where the two predictions differ.

	#Diff.	corr.	new err.	changed err.
0	45'552	34.22%	62.59%	3.19%
,	50'496	69.01%	28.30%	2.69%
.	16'190	49.28%	44.69%	6.03%
-	4'422	51.15%	33.04%	15.81%
:	2'014	41.46%	31.43%	27.11%
?	1'158	63.90%	29.53%	6.56%

Table 9: Detailed comparison of the differences in Unbabel's predictions versus OnPoint's predictions for English in subtask 2. *#Diff.* signifies the number of tokens that have the respective label as the ground truth and for which OnPoint's and Unbabel's predictions differ. The remaining columns represent the percentage of this number in each difference class.

In conclusion, we observer that while the top three systems perform similarly in terms of Macro F1 scores for subtask 2, there are nuances to each system that distinguishes them from the others.

5.1 Winners

While we showed that there are differences in the outputs of the top three systems that are not reflected in the averaged F1 scores, the declared criteria for winning the task are the averaged F1 scores in Tables 4 and 5. Since the top three systems in these tables are practically indistinguishable based on these F+ scores, we declare OnPoint, htw+t2k, and Unbabel as the joint winners of the SEPP-NLG 2021 shared task. Congratulations!

6 Conclusions

We presented the setting and results of the first Sentence End and Punctuation Prediction in NLG text (SEPP-NLG 2021) shared task. We found that all participants explored neural networks-based models (particularly transformers) to tackle the task. The results for the in-domain Europarl data were high for the most common punctuation symbols, but the performance decreased significantly when the models were faced with out-of-domain data.

The discussion of the task results during the session at the SwissText conference yielded the following desiderata for future iterations of the shared task:

- More heterogeneous data (more domains)
- Add truecasing as an additional task
- Add other language families
- Take inference time / computational costs as an additional evaluation criteria, or create a separate track that puts emphasis on a low-resource/low-latency setting

Acknowledgments

We thank the participants for their submissions and their valuable feedback on early versions of the data and task details. This work was funded by Innosuisse under grant project nr. 43446.1 IP-ICT.

References

Eunah Cho, Jan Niehues, and Alex Waibel. 2017. Nmt-based segmentation and punctuation insertion for real-time spoken language translation. In *Interspeech*, pages 2645–2649.

Nn. Deepamala and P. Ramakanth. 2012. Sentence boundary detection in kannada language. *International Journal of Computer Applications*, 39:38–41.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Gregor Donabauer and Udo Kruschwitz. 2021. University of regensburg @ swisstext 2021 sepp-nlg: Adding sentence structure to unpunctuated text. In *Proceedings of the 1st Shared Task on Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG 2021) at SwissText 2021*.

Gregor Donabauer, Udo Kruschwitz, and David Corney. 2021. Making sense of subtitles: Sentence boundary detection and speaker change detection in unpunctuated texts. In *Companion Proceedings of the Web Conference 2021*, pages 357–362.

Gregory Grefenstette and Pasi Tapanainen. 1997. What is a word, what is a sentence? problems of tokenization.

Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans-Joachim Bohme. 2021. Fullstop: Multilingual deep models for punctuation prediction. In *Proceedings of the 1st Shared Task on Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG 2021) at SwissText 2021*.

Jagroop Kaur and Jaswinder Singh. 2019. Deep neural network based sentence boundary detection and end marker suggestion for social media text. In *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 292–295.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Comput. Linguist.*, 32(4):485–525.

Ondřej Klejch, Peter Bell, and Steve Renals. 2017. Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5700–5704. IEEE.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Machine Translation Summit, 2005*, pages 79–86.

Roque López and Thiago A. S. Pardo. 2015. Experiments on sentence boundary detection in user-generated web content. In *Computational Linguistics and Intelligent Text Processing*, pages 227–237, Cham. Springer International Publishing.

Jose Manuel Masiello-Ruiz, Jose Luis Lopez Cuadrado, and Paloma Martinez. 2021. Participation of hulat-uc3m in sepp-nlg 2021 shared task. In *Proceedings of the 1st Shared Task on Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG 2021) at SwissText 2021*.

Andrianos Michail, Silvan Wehrli, and Terézia Bucková. 2021. Uzh onpoint at swisstext-2021: Sentence end and punctuation prediction in nlg text through ensembling of different transformers. In *Proceedings of the 1st Shared Task on Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG 2021) at SwissText 2021*.

Vandan Mujadia, Pruthwik Mishra Dipti, and Misra Sharma. 2021. Deep contextual punctuator for nlg text. In *Proceedings of the 1st Shared Task on Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG 2021) at SwissText 2021*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Ines Rehbein, Josef Ruppenhofer, and Thomas Schmidt. 2020. Improving sentence boundary detection for spoken language transcripts. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC), May 11-16, 2020, Palais du Pharo, Marseille, France*, pages 7102–7111. European Language Resources Association.

Ricardo Rei, Fernando Batista, , Nuno M. Guerreiro, and Luisa Coheur. 2021. Multilingual simultaneous sentence end and punctuation prediction. In *Proceedings of the 1st Shared Task on Sentence End and Punctuation Prediction in NLG Text (SEPP-NLG 2021) at SwissText 2021*.

Ricardo Rei, Nuno Miguel Guerreiro, and Fernando Batista. 2020. Automatic truecasing of video subtitles using bert: A multilingual adaptable approach. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 708–721, Cham. Springer International Publishing.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. ANLC '97, page 16–19, USA. Association for Computational Linguistics.

Michael D. Riley. 1989. Some applications of tree-based modelling to speech and language. HLT '89, page 339–352, USA. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ottokar Tilk and Tanel Alumäe. 2016. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *INTERSPEECH*.

Don Tuggener. 2017. A method for in-depth comparative evaluation: How (dis)similar are outputs of pos taggers, dependency parsers and coreference resolvers really? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 188–198, Valencia, Spain. Association for Computational Linguistics.

Daniel J. Walker, David E. Clements, Maki Darwin, and Jan W. Amtrup. 2001. Sentence boundary detection: A comparison of paradigms for improving mt quality. In *In Proceedings of MT Summit VIII: Santiago de Compostela*, pages 18–22.

Jiangyan Yi and Jianhua Tao. 2019. Self-attention based model for punctuation prediction using word and speech embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7270–7274. IEEE.

Piotr Żelasko, Piotr Szymański, Jan Mizgajski, Adrian Szymczak, Yishay Carmiel, and Najim Dehak. 2018. Punctuation prediction model for conversational speech. *Proc. Interspeech 2018*, pages 2633–2637.